

## **REMARKS**

The enclosed is responsive to the Examiner's Office Action mailed on July 27, 2006. At the time the Examiner mailed the Office Action claims 126-142 were pending. By way of the present response the Applicants have: 1) amended claims 126, 130, 132, 137, and 141; 2) added new claims 143-167; and 3) canceled no claims. As such, claims 126-167 are now pending. The Applicants respectfully request reconsideration of the present application and the allowance of all claims now presented.

### **Claim Rejections Under 35 U.S.C. § 103(a)**

Claims 126-142 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Holt, et al., U.S. Patent No. 5,557,723 ("Holt"), in view of Hitchcock, et al., U.S. Patent No. 6,460,042 ("Hitchcock"), and in view of Texier, U.S. Patent No. 5,119,476 ("Texier").

Holt describes a "method and system for user-customizable forms in an electronic mail system." (Holt, Abstract; *see, also*, Fig. 3 illustrating an exemplary form.) A user in Holt's system may "specify the layout of a custom form and specify a for control procedure (FCP) to control the behavior of the form components. The FCP is a computer subroutine that implements user-defined processing of the form." (Holt, col. 2, lines 46-51.) "The FCP is written by the forms designer to implement form customization." (Holt, col. 5, lines 56-57.)

Hitchcock describes the creation of customizable forms and sharing between customizable forms. (Hitchcock, Abstract.) A user "opens an account" that allows user inputted data (such name, address, etc.) to stored in a database and shared between forms.

(Hitchcock, Abstract.) “When the applicant requests an application form for a particular institution ... forms engine 104 retrieves user information regarding the status of applications that are pending or completed. Forms engine 104 then generates a customized application form based upon an application description in an application data file 108. Forms engine 104 then retrieves user data that was entered in previous applications and stored in the applicant database 62, and merges the user data into the current application, which is then returned to the applicant as an HTML form.”

(Hitchcock, col. 5, line 57 to col. 6, line 2.) The application data file “is a series of ‘directives’ and optional arguments which the forms engine parses to build the HTML form and to merge in user data.” (Hitchcock, col. 10, lines 43-45.)

Texier describes a “method of creating data entry forms.” (Texier, Abstract.) “The invention offers a method for generating dialogue windows, hereafter called forms, which can be displayed on the screen of a computer system...” (Texier, col. 1, lines 61-63.) Each window may utilize a set of reactive functions (an “editor”) that are “responsive to activation by one or more external events” in an “active zone.” (Texier, col. 2, lines 8-15; see, also, col. 8, line 23 to col. 9, line 62 for editor functions.) New form creation is illustrated in Fig. 2 and its associated text. Fig. 3 shows a form (“foto:1”) and its associated information (the editor function being used “LS”, font, etc.). Texier describes the manual creation of a form.

With respect to claims 126 and 137, the combination of Holt, Hitchcock, and Texier does not describe what Applicant’s claim requires. Specifically, the combination does not describe:

receiving, through a network, a form authored using  
a form authoring language, the form containing one or  
more input fields;

parsing the received form to identify the input fields contained in the received form;  
providing a graphical user interface to allow identification of actions to be associated with the identified input fields upon subsequent specific submission of a specific instance the form by a third party, the provided graphical user interface being dependent on the identified input fields;  
automatically generating a program code to carry out the actions associated with the identified input fields.

Each of the references describes the generation of a form. While Hitchcock uses an application data file to store “a series of directives” which a forms engine parses this parsing is only to build a new HTML form and to merge in user data. None of the references, alone or in combination, describe receiving or parsing an existing form to identify input fields of the form. To the contrary, each reference describes the generation of a form.

None of the references, alone or in combination, describe automatically generating program code to carry out action associated with identified input fields of an existing form. The FCP of Holt requires a programmer to create any sort of action to be taken. Holt further demonstrates that Holt expressly teaches away from “automatically generating a program code to carry out the actions identified by the user,” as discussed above. The user must write the code him/herself in Holt. Holt merely states that the user can specify an FCP to control the behavior of the form components. (See, for example, Holt, col. 2, lines 44-57.) Thus, as set forth further above, Holt nowhere teaches automatic generation of the FCPs. The other references do not address the generation of program code.

Applicant’s claim is not directed toward the authoring of a form. Applicant’s claim requires “receiving a form,” “parsing the received form to identify...input

fields...in the form,” providing a GUI “to allow identification of actions to be associated with the ...input fields,” and “automatically generating a program code to carry out the actions associated with the identified input fields.” Applicant’s claim does not require the generation of a new form or even the modification of the form itself as described by Holt, Hitchcock, and Texier individually or in combination. Additionally, the references, as a whole, are incompatible and are not combinable.

Applicant respectfully submits that claims 126 and 137 are not described by the combination of Holt, Hitchcock, and Texier. Claims 127-135 are dependent upon claim 126 and claims 138-142 are dependent upon claim 137 and are allowable for at least the same reasons as claims 126 and 137 respectively.

With respect to claim 132, the combination of Holt, Hitchcock, and Texier does not describe what Applicant’s claim requires. Specifically, the combination does not describe:

A server comprising:

a communications device connected to a network to receive a form from a client connected to the network, the form authored using a form authoring language and containing one or more input fields;

a memory coupled to the communications device to store the received form;

a parser module coupled to the memory to parse the received form to identify the input fields contained in the received form;

a configurer module coupled to the parser module to create a graphical user interface based on the input fields identified by the parser module, and to provide the graphical user interface using the communications device to allow the identification of actions to be associated with the identified input fields upon subsequent submission of a specific instance of the form by a third party;

a code generator module coupled to the configurer module to automatically generate a program code to carry out the actions identified.

Each of the references describes the generation of a form. While Hitchcock uses an application data file to store “a series of directives” which a forms engine parses this parsing is only to build a new HTML form and to merge in user data. None of the references, alone or in combination, describe receiving or parsing an existing form to identify input fields of the form. To the contrary, each reference describes the generation of a form.

None of the references, alone or in combination, describe automatically generating program code to carry out action associated with identified input fields of an existing form. The FCP of Holt requires a programmer to create any sort of action to be taken. Holt further demonstrates that Holt expressly teaches away from “automatically generating a program code to carry out the actions identified by the user,” as discussed above. The user must write the code him/herself in Holt. Holt merely states that the user can specify an FCP to control the behavior of the form components. (See, for example, Holt, col. 2, lines 44-57.) Thus, as set forth further above, Holt nowhere teaches automatic generation of the FCPs. The other references do not address the generation of program code.

Applicant’s claim is not directed toward the authoring of a form. Applicant’s claim requires a communications device to “receive a form,” a parser module to “parse the received form to identify...input fields...in the form,” a GUI “to allow identification of actions to be associated with the ... input fields,” and a code generator to “automatically generating a program code to carry out the actions the identified.”

Applicant's claim does not require the generation of a new form or even the modification of the form itself as described by Holt, Hitchcock, and Texier individually or in combination. Additionally, the references, as a whole, are incompatible and are not combinable.

Applicant respectfully submits that claim 132 is not described by the combination of Holt, Hitchcock, and Texier. Claims 127-135 are dependent upon claim 126 and are allowable for at least the same reasons.

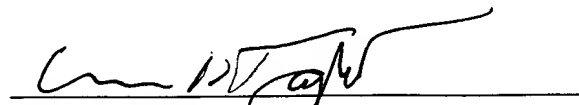
Therefore, Applicant respectfully submits that all claims presently pending are allowable.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due. Furthermore, if an extension is required, then Applicants hereby request such an extension.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: 1/29/07



Edwin H. Taylor  
Reg. No.: 25,129

12400 Wilshire Boulevard  
Seventh Floor  
Los Angeles, California 90025-1030  
(408) 720-8300